LETTER **Communicated by Neil Burgess**

A Principle for Learning Egocentric-Allocentric Transformation

Patrick Byrne
pbyrne@yorku.ca
Suzanna Becker
becker@mcmaster.ca
Department of Psychology, Neuroscience and Behavior, McMaster University, Hamilton, Ontario, L8S 4K1, Canada

Numerous single-unit recording studies have found mammalian hippocampal neurons that fire selectively for the animal's location in space, independent of its orientation. The population of such neurons, commonly known as place cells, is thought to maintain an allocentric, or orientation-independent, internal representation of the animal's location in space, as well as mediating long-term storage of spatial memories. The fact that spatial information from the environment must reach the brain via sensory receptors in an inherently egocentric, or viewpoint-dependent, fashion leads to the question of how the brain learns to transform egocentric sensory representations into allocentric ones for long-term memory storage. Additionally, if these long-term memory representations of space are to be useful in guiding motor behavior, then the reverse transformation, from allocentric to egocentric coordinates, must also be learned. We propose that orientation-invariant representations can be learned by neural circuits that follow two learning principles: minimization of reconstruction error and maximization of representational temporal inertia. Two different neural network models are presented that adhere to these learning principles, the first by direct optimization through gradient descent and the second using a more biologically realistic circuit based on the restricted Boltzmann machine (Hinton, 2002; Smolensky, 1986). Both models lead to orientation-invariant representations, with the latter demonstrating place-cell-like responses when trained on a linear track environment.

1 Introduction

Animals are often faced with the challenging task of deciding how to act in the absence of complete sensory information, for example, when navigating toward an unseen goal. Instead they must rely on internal representations of object locations within their environment. From numerous electrophysiological and behavioral studies, it has become clear that space is represented in multiple reference frames across many different brain regions. Such representations can be divided into two general classes: allocentric or orientation-independent representations and egocentric representations. For example, orientation-invariant hippocampal place cells fire selectively for a rat's location in space (e.g., O'Keefe, 1976) but show little dependence on the animal's orientation in the open field (Muller, Bostock, Taube, & Kubie, 1994). Place cells have also been found in the hippocampus of nonhuman primates (Matsumura et al., 1999; Ono, Nakamura, Nishijo, & Eifuku, 1993) and humans (Ekstrom et al., 2003). More recently, orientation-invariant grid cells, which fire selectively at the vertices of a triangular lattice covering the environment, have been found in the medial entorhinal cortex of rats (Hafting, Fyhn, Molden, Moser, & Moser, 2005; Sargolini et al., 2006). In addition, representations of orientation independent of location have been found in head direction cells (see, e.g., Taube, 1998), which are located in various brain regions including the anterior thalamus. Finally, viewcells, which fire when an animal is looking at a given object or location from a range of vantage points, have also been found in the hippocampal region in both nonhuman (Matsumura et al., 1999; Rolls & O'Mara, 1995) and human primates (Ekstrom et al., 2003). Furthermore, numerous studies on human and nonhuman animals with hippocampal or medial temporal lobe (MTL) lesions have left little doubt that these structures are critical for performing a variety of spatial memory tasks, especially those involving viewpoint changes, and after medium to long time delays between encoding and retrieval (see, e.g., Barnes, 1988; Bohbot et al., 1998; Crane & Milner, 2005; Jarrard, 1993; King, Burgess, Hartley, Vargha-Khadem, & O'Keefe, 2002; Morris, Garrard, Rawlins, & O'Keefe, 1982; see Burgess, Maguire, & O'Keefe, 2002, for a review).

Unlike allocentric representations of space, which are supported primarily by MTL structures, egocentric representations are common in various neocortical regions, including motor and sensory cortices. For example, the posterior parietal cortices appear to represent the locations of objects in combinations of reference frames, such as retinotopic modulated by head direction (Zipser & Andersen, 1988) or retinotopic modulated by body direction within the room (Snyder, Grieve, Brotchie, & Andersen, 1998). The relative contribution of egocentric and allocentric representations to spatial memory might depend on the timescale of the task concerned (see, e.g., Milner, Paulignan, Dijkerman, Michel, & Jeannerod, 1999). Short-term retention of perceptual information for the purpose of immediate action will be best served by egocentric representations appropriate to the corresponding sensory and motor systems. By contrast, long-term memory for locations will be best served by allocentric representations because the location and configuration of the body during retrieval generally will not be the same as during encoding (see Burgess, Becker, King, & O'Keefe, 2001, for further discussion). This view is consistent with single unit recording results from monkey dorsolateral prefrontal and posterior parietal cortices that suggest spatial working memory is indeed egocentric in nature (Chafee &

Goldman-Rakic, 1998; Funahashi, Bruce, & Goldman-Rakic, 1989), whereas medium- to long-term spatial representations in the MTL are allocentric, as described above.

In summary, long-term spatial memory seems to rely heavily on allocentric MTL representations, while short-term memory may consist of egocentric representations maintained in frontoparietal cortical circuits, with parietal cortex maintaining representations in multiple frames. The existence of egocentric representations is not surprising given that sensory information arrives at the brain in an intrinsically viewpoint- or body-configuration-dependent fashion. However, allocentric representations must arise indirectly via transformation of egocentric sensory or internal representations. Elsewhere (Becker & Burgess, 2001; Byrne, Becker, & Burgess, 2007; Byrne & Becker, 2004), we have presented neural circuit models that accomplish such transformations. However, an important question not addressed in our previous work is how such transformations can be learned. Numerous neural network models have addressed how coordinate transformations might be accomplished in parietal cortex, (see, e.g., Pouget & Sejnowski, 1997; Salinas & Abbott, 1996); however, not all of these models address learning, and when they do, they focus on transformation from one type of egocentric reference frame to another (see, e.g., Deneve, Latham, & Pouget, 2001; Mazzoni, Andersen, & Jordan, 1991; Xing & Andersen, 2000; Zipser & Andersen, 1988). Other neural models have attempted to address the learning of place cell responses from purely egocentric input (see, e.g., Sharp, 1991), or from spatial geometric information already transformed partially into an allocentric frame (see, e.g., Kali & Dayan, 2000). Trullier, Wiener, Berthoz, and Meyer (1997) has reviewed a number of navigation models, many of which rely on a transformation from egocentric to allocentric representations of space. As with other models of place cell learning, none of these learns the full inverse transformation that would allow the entire egocentric geometry of an environment to be recovered from the corresponding allocentric representation. This inverse transformation is essential for mental imagery of retrieved memories and long-term-memory-guided navigation. We now propose two simple principles that may underlie the learning of full egocentric-allocentric transformation ability. The issue of how the orientation-invariant representations generated by the application of these principles might relate to grid cells and place cells will be addressed briefly in the discussion.

The first principle, which we refer to as the *minimization of reconstruction error*, is based on the observation that if the brain transforms an egocentric representation of space into an allocentric one for long-term storage, then it must be able to recover the original egocentric information from the stored allocentric information in order to drive sensory-motor circuits. The second principle, which we refer to as the *maximization of temporal inertia*, is motivated by electrophysiological evidence for neurons with sustained response properties in various MTL structures. For example, Redish, McNaughton, and Barnes (2000) found that when rats shuttled back and forth along a linear track, many place cells were directionally tuned and either ceased or began firing after a change of direction; moreover, those that began firing after a direction change did so abruptly, while those that ceased firing did so more gradually, exhibiting inertia. Further evidence for temporal inertia comes from unit recordings performed in slice preparations of rat entorhinal cortex. When bathed in carbachol, an acetylcholine receptor agonist, neurons in deep layers of entorhinal cortex demonstrated strong resistance to changes in their firing rates (Egorov, Hanmam, Fransen, Hasselmo, & Alonso, 2002). This finding is particularly relevant given the evidence that encoding processes in MTL structures may be facilitated by elevated levels of acetylcholine (for an excellent review, see Hasselmo, 1999). Thus, we suggest that MTL representations vary as slowly as possible in time.

We now describe a set of simulations of neural networks with a simple architecture that can learn to generate orientation-independent representations of simple environments when learning is based on the two principles presented above. The first model we describe is trained using learning equations derived by direct minimization of a cost function based on these principles. The purpose of simulations with the first model was to show that the two learning principles were sufficiently constraining to result in orientation-independent representations. We then describe a second neural network model that is trained using more biologically plausible, contrastive Hebbian learning in a restricted Boltzmann machine (RBM) architecture (Hinton, 2002). The learning rules for training an RBM also have a clear statistical interpretation that is roughly equivalent to our first principle of minimizing reconstruction error. By adding a temporal inertia constraint to this model, we hypothesized that orientation-independent representations of space would emerge.

2 Simulation 1: Cost Function Method Applied to a Simple Linear Environment

2.1 Methods.

2.1.1 Model Architecture and Neural Dynamics. The model architecture, depicted in Figure 1, is based on a transformation circuit presented by Byrne et al. (2007). The input layer of neurons maintains a continuously updated egocentric representation of the environmental geometry about the model during simulated navigation. This spatial information is represented in a head-centered reference frame, in which the model's location and head direction are always fixed. Given this choice of reference frame and additional input from a population of head direction cells, the transformation from an egocentric to an orientation-independent reference frame can be

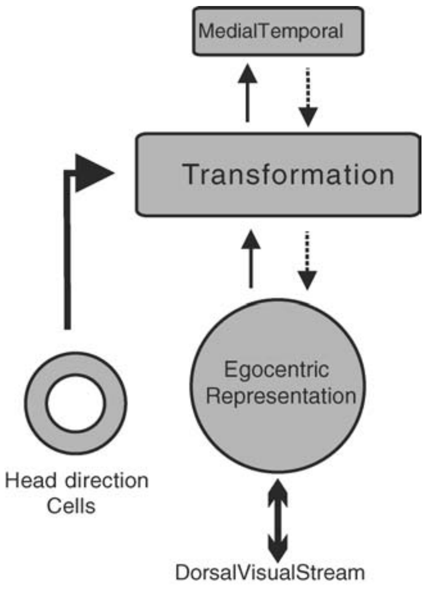

Figure 1: Schematic of the overall model architecture. During a bottom-up cycle, all connections represented by solid black arrows are active. Thus, head-centered egocentric representations are driven by input from the dorsal visual stream that, along with head direction cell input, drives medial temporal activity via a transformation layer. During a top-down cycle, connections represented by vertical solid arrows become downregulated while connections represented by dashed arrows become upregulated. Thus, in this phase, egocentric representations are driven by MTL and head direction activity.

accomplished. We assume that such a transformation must take place in the brain, based on the evidence for head direction cells, place cells, grid cells, and the numerous egocentric reference frames in which representations of space are found to exist in posterior parietal and frontal cortical areas.

The head-centered egocentric representation of the environment, combined with the activity of the head direction cells, provides the information required to transform this representation into an orientation-independent one. Hence, our model contains a layer of transformation neurons that are driven by the combined activity of the head direction cells and the egocentric layer neurons. Activity from the transformation layer feeds forward to a layer of MTL neurons that should, in principle, be able to learn orientation-invariant representations. For neurons in all layers except the egocentric input layer, the firing rate of the ith neuron in layer α is a sigmoid function

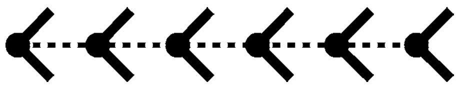

Figure 2: Schematic of the very simple linear environment. Each possible traversed location is represented by a filled circle, while each possible head direction at a given location is represented by a solid line segment emanating from the circle representing that location. The dashed line indicates the path followed by the model from left to right.

of the summed weighted activities of the presynaptic neurons;

$$R_i^\alpha = \frac{1}{1 + \mathrm{e}^{-\sum_j w_{ij}^\beta R_j^\beta}}, \tag{2.1}$$

where α represents either the transformation layer or the MTL layer; β represents an incoming weight from either (1) the head direction cell layer and the egocentric layer or (2) the transformation layer, depending on the value of α; w_{ij}^β is the connection strength from neuron j in layer β to neuron i in layer α; and R_j^β is the firing rate of neuron j in layer β. In this simulation, rate-coded neurons are employed, and the firing rate, R, is an abstract representation of how much activity a neuron exhibits based on presynaptic activity. In simulations 2 and 3, binary stochastic neurons are employed, and this firing rate will be replaced with a firing state, S, which takes on a value of one or zero, depending on whether a neuron fires on a given time step.

So far, we have discussed how the model operates in bottom-up mode, whereby incoming sensory information drives egocentric representations that in turn drive representations in the MTL component of the model. In addition, the model can operate in a top-down fashion in which MTL and head direction cell activity drive activity in the egocentric layer via the transformation layer. Thus, after learning, the model should be able to reconstruct egocentric representations of space from allocentric representations in long-term memory, given a particular head direction.

2.1.2 Cost Function Minimization. In simulation 1, the model was trained on the simple, abstract linear environment depicted in Figure 2. This environment consisted of six locations, each with two head directions. The 12 head-centered egocentric representation vectors, one for each conjunction of location and head direction (this conjunction will be referred to as a *viewpoint* from this point on), were chosen to be four-element, real-valued vectors with the correlational structure shown in the top panel of Figure 3. Specifically, they were chosen so that the overlap between vectors (measured as cosine of angle between vectors) corresponding to nearby

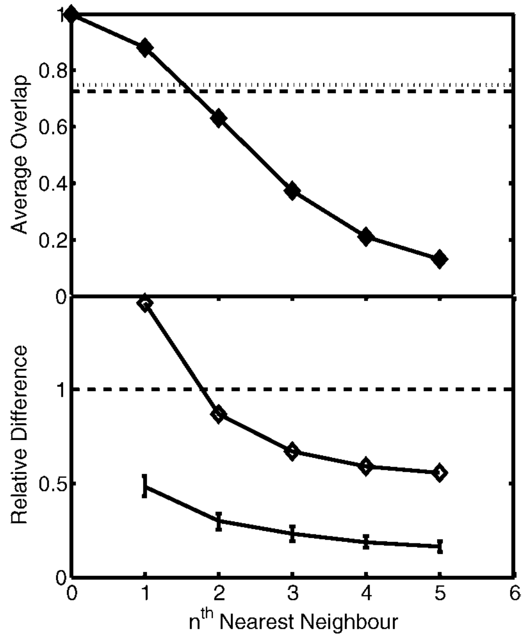

Figure 3: (Top) The filled diamonds represent the average overlap between pairs of (four-element, real-valued) egocentric representation vectors for the very simple linear environment when separated by n neighbors through pure translation; plotted as a function of n. Overlap is measured as the cosine of the angle between the vectors. The dotted line represents the average overlap between pairs of random vectors, while the dashed line represents the average overlap between pairs of egocentric representation vectors differing in rotation at the same location. (Bottom) Orientation-invariance results for the model trained to minimize the cost function given by equation 2.2 on the simple linear track environment. The diamonds are $D_E(n)$, our measure of orientation invariance for egocentric representations, while the curve with error bars is $D_M(n)$, our measure of orientation invariance for MTL representations. Error bars represent standard error of the mean calculated across ten networks trained with random initial conditions.

locations with the same head direction was large and fell off with increasing translational separation, while overlap between vectors corresponding to the same location but different headings had a constant, intermediate value of just under 0.75, regardless of location. Also, for this simulation, the head direction variable could take on one of only two values, so that the model's current head direction was represented by the firing of one of two head direction neurons. We do not consider what drives the head direction system, but we do assume that it is reliable in the sense that it will orient the model consistently within a particular environment whenever it is placed in the same location with the same input cues.

A constant speed trajectory through the environment can be represented by the ordered set of coordinates, $\tau = \{x_i\}_{i=1}^{N}$, where x_i represents the conjunction of the model's position coordinates and heading direction (i.e., viewpoint) at discrete time, i. Our two learning principles can now be incorporated explicitly into the following cost function,

$$C(\tau) = \sum_{i=1}^{N} \left\{ [E(x_i) - E_r(x_i)]^2 + \left[\frac{M(x_i) - M(x_{i-1})}{M(x_i)} \right]^2 \right\}, \tag{2.2}$$

where $E(x)$ is the egocentric representation vector at x, E_r is the model's reconstruction of E, and $M(x)$ is the MTL representation generated by the model at x. Here, reconstruction error is penalized directly by the first term on the right-hand side, while the second term penalizes changes in MTL activity from one point along the trajectory to the next.

In simulation 1, the model was trained on the simple linear environment by performing gradient descent on the cost function given by equation 2.2 as the model navigated from the left-most position to the right-most position of the simple linear environment. Navigation within this environment consisted of steps that alternated between those in which the model hopped from its current location to the rightward neighboring location while its head direction was randomized, and those in which it maintained its current location and switched head directions. This process resulted in 64 possible unique left-to-right trajectories through the environment. Each term in the summation of equation 2.2 corresponds to one step in the navigation process and was calculated immediately after one bottom-up followed by one top-down cycle was performed at that step's corresponding location; at the same time, the gradient of that term was calculated analytically. Although weights could have been updated in an online fashion after each such navigation step, we found that this approach led to large fluctuations in the cost function. Instead, the total gradient of equation 2.2 was calculated by accumulating the individual gradients over five full left-to-right traversals of the environment before performing a single weight update step. Weight updates were performed based on the total gradient using the RPROP algorithm (Riedmiller & Braun, 1993), a more efficient version of the well-known backpropagation algorithm (LeCun, 1985; Parker, 1985; Rumelhart, Hinton, & Williams, 1986; Werbos, 1974), with parameters $\eta^{+} = 1.2$, $\eta^{-} = 0.5$, $\Delta_{\max} = 1$, and $\Delta_0 = 0.001$. The RPROP algorithm employs a variable step size for each component of the gradient, with $\eta^{+/-}$ determining how much the step size for a given component should increase or decrease from one step to the next, with $\Delta_{\max}$ being the largest possible step size for any component, and with Δ_0 being the initial step size for all components.

2.1.3 Data Analysis. To evaluate the effectiveness of the learning algorithm in generating allocentric representations, a measure of orientation invariance for MTL representations was required. We chose a measure of how different, on average, MTL representations are from each other when separated by a pure rotation, as compared with how different they are on average when separated by a pure translation step through n neighbors. This measure will be low, indicating allocentricity, if representations vary relatively little across rotations at a given location and relatively more across neighboring locations. Although locations within the simple linear environment were not associated with specific Cartesian coordinates, locations within the environments used for simulations 2 and 3 were. Moreover, the environment used for simulation 3 was two-dimensional. Therefore, the measure of orientation invariance is presented now in the general form. It is given by

$$D_M(n) = \sqrt{\frac{\langle [M(\mathbf{r},\theta) - M(\mathbf{r},\theta')]^2 \rangle_{\mathbf{r},\theta,\theta'}}{\langle [M(\mathbf{r},\theta) - M(\mathbf{r}+\mathbf{n},\theta)]^2 \rangle_{\mathbf{r},\theta,\hat{\mathbf{n}}}}}, \tag{2.3}$$

where M is a vector of activations in the MTL region of the model, $\mathbf{r}$ is the vector of Cartesian coordinates of an environmental grid point, θ and θ' are two unequal head directions, $\mathbf{n}$ is a vector from $\mathbf{r}$ to an nth nearest neighbor ($n = 1$ for nearest neighbor, 2 for next nearest, and so on), $\hat{\mathbf{n}}$ is a unit vector pointing to a neighbor, and $\langle \ldots \rangle$ indicates averaging with respect to the subscripts. For orientation-invariant representations, $D_M(n)$ should be much smaller than one, especially for large values of n. As a basis for comparison, the relative difference measure of equation 2.3 can be applied to the various egocentric representations of the environment as well. We denote the latter case by $D_E(n)$ and expect that orientation-invariant MTL representations would also satisfy $D_M(n) \ll D_E(n)$. For the current environment, each average in equation 2.3 was calculated with 5000 random samples.

2.2 Results. Initially the egocentric-to-transformation layer and transformation-to-MTL layer weights were set to random values between −0.2 and 0.2, while head direction-to-transformation layer weights were set to random values between −2 and 2. These values were chosen because they were found to generate reasonable levels of activity in all layers when the model was presented with egocentric and head direction input. The initial head direction-to-transformation layer weights were chosen from a larger range than the other weights so that the activity of head direction neurons, which were fewer in number than were the egocentric neurons, could have an influence on transformation layer activity comparable to that of activity from the egocentric layer.

Orientation-invariance results averaged over 10 sets of random initial conditions for a network with eight transformation units and four MTL units are shown in the bottom panel of Figure 3. Notice that D_M is small compared to D_E and is also much smaller than one. Therefore, MTL representations are much more orientation-invariant than the representations at the egocentric input layer.

2.3 Larger Environments. In preliminary simulations we used the cost function minimization approach to train the transformation model on a more complex environment than that used for simulation 1. Egocentric representations for this environment were derived directly from the configuration of rectilinear boundaries that specified its geometry (a set of three interconnected "rooms") using a process described in the next section. Starting from random initial weights chosen from reasonable ranges, training tended to lead to local minima in which MTL representations did not demonstrate orientation invariance. Performing the minimization as described above was also costly in terms of time, and we therefore did not attempt any computationally intensive global optimization procedures (annealing, for example). Instead, we were able to show that weights that were precalculated to generate orientation invariance in MTL representations (taken from Byrne et al., 2007) represent a local minima of the cost function, thereby demonstrating that our principles are consistent with the learning of such representations.

3 Simulation 2: RBM Method Applied to Linear Environment

3.1 Methods. Training our transformation circuit by following the negative gradient of equation 2.2 is problematic in at least two ways. First, the learning rules resemble those of backpropagation and thus do not resemble any biological process known to occur in the brain. Second, it appears that the learning is highly susceptible to becoming trapped in poor local minima in all but the simplest of abstract environments. In this section we demonstrate that training the circuit as two "stacked" restricted Boltzmann machines alleviates these problems.

3.1.1 RBM Learning Rule. A restricted Boltzmann machine, or RBM (Smolensky, 1986) is a special case of the Boltzmann machine (Ackley, Hinton, & Sejnowski, 1985) with two interconnected layers and no within-layer connections. We use two stacked layers of RBMs here (Hinton, Osindero, & Teh, 2006) with stochastic binary units, each of which can take on a value, or firing state, of zero or one (see Figure 4). In such a network, a neuron in the visible layer is connected to all neurons in the hidden layer but to none of the other visible neurons. Similarly, a neuron in the hidden layer is connected to all visible units but to none of the hidden units. Weights in the network are symmetric so that the connection strength from unit i to

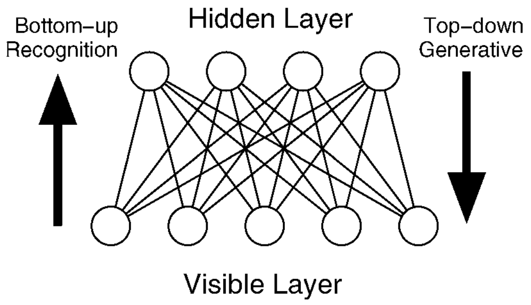

Figure 4: Schematic of an RBM network. In a bottom-up cycle, data are clamped onto visible units, and a hidden state is generated. In a top-down cycle, a hidden state is clamped, and a visible state is generated.

unit j is equal in value to the connection strength from unit j to unit i. In terms of dynamics, the probability that a given neuron will fire at a given time step is determined by the firing state of all of the other neurons at the previous time step. In particular, this probability is a sigmoid function of the weighted summed input to the neuron and is given by

$$p_i^\alpha = \frac{1}{1 + e^{-\sum_j w_{ij}^\beta S_j^\beta}}, \tag{3.1}$$

where all symbols have the same meaning as in equation 2.1, except that the firing rate R has been replaced with firing probability p on the left-hand side and with firing state S on the right-hand side. Since there are no intralayer connections in an RBM, hidden activity is determined solely by visible activity and vice versa. Thus, given the firing states of one layer, the probabilities of each of the units firing in the other layer are conditionally independent, and the joint probability of the entire layer can therefore be calculated in a single step. If this alternating process is continued, then a global equilibrium state will eventually be reached. Hinton (2002) derived learning rules for this network such that the trained network, under the repeated alternating Gibbs sampling procedure just described, will produce visible states with probabilities similar to that with which they occur in the training set. Moreover, the learning rules can be shown to minimize a term closely related to reconstruction error (Hinton, 2002). Thus, if a data vector is clamped onto the visible units after training and a hidden state is calculated from it, then clamping this hidden state will generate a visible state that is highly similar to the original data vector. Although RBMs resemble another neural network architecture, the bidirectional associative memory (BAM) network of Kosko (1988), the former "discover" their own nonunique internal (hidden) representations of data while the latter heteroassociative networks must be trained with fully specified pairs of neural activation vectors. As will become clear, the flexible nature of RBM learning is necessary for our current purposes.

The RBM learning rule involves a two-phase learning procedure and consists of the weight update equation given by

$$\Delta w_{ij} = \eta(\langle S_i S_j \rangle_{\mathbf{d}} - \langle S_i S_j \rangle_{\hat{\mathbf{d}}}), \tag{3.2}$$

where w_{ij} is the weight connecting neurons i and j, η is a learning rate parameter, S_i is the firing state of the ith neuron $\langle \ldots \rangle_{\mathbf{d}}$ represents averaging over the data distribution in the Hebbian phase, and $\langle \ldots \rangle_{\hat{\mathbf{d}}}$ represents averaging over one-step reconstructions obtained in the anti-Hebbian phase (explained below). The Hebbian phase correlation is calculated by repeatedly clamping the states of the visible units to a data vector randomly sampled from the data set, calculating a corresponding hidden state, and averaging the resulting correlations. Starting from the hidden unit state obtained in the Hebbian phase, the anti-Hebbian phase states are obtained by updating the states of the visible units once to obtain data reconstructions and then updating the hidden units once. The resulting correlations are obtained by averaging across states obtained in this manner from one-step-away reconstructions of data vectors.

In practice, rather than averaging across many data vectors, equation 3.2 can be implemented online by updating the weights after each sample, with Hebbian updates interleaved with anti-Hebbian updates. Unless an RBM has learned a data distribution perfectly, it is clear that a data vector reconstructed by the network will be contaminated with preexisting correlations arising from the original untrained weights, as well as correlations due to other subsequently learned data vectors. Thus, the stochastic Hebbian updates of the RBM learning procedure build data correlations into the network, while the anti-Hebbian updates remove interfering correlations.

3.1.2 Acetylcholine and Temporal Inertia. Hasselmo and McGaughy (2004) provide a detailed review of evidence suggesting that acetylcholine might mediate a two-phase learning process in the brain that strongly parallels the RBM rules. In particular, when acetylcholine levels are high, as observed in active wakefulness, neural activity in neocortical and MTL areas is influenced strongly by sensory afferents, and long-term potentiation is facilitated in various hippocampal structures. However, when acetylcholine levels are low, as observed during quiet wakefulness or non-REM sleep, the influence of sensory input on neocortical and MTL activity is reduced relative to the influence of feedback processes. Thus, Hasselmo and McGaughy suggest that these high and low acetylcholine states correspond to bottom-up learning and top-down consolidation, respectively. Consistent with these ideas, Foster and Wilson (2006) have found that place cell activity recorded as a rat traverses a linear track immediately replays itself in reverse when the animal rests at the end of the track.

Interestingly, conditions of high acetylcholine levels appear to correspond to the electrophysiological findings of high temporal inertia as discussed in section 1. In particular, the inertia-like processes in entorhinal cortex slice preparations (Egorov et al., 2002) were observed in the presence of carbachol, while the inertia-like neuronal properties in rat hippocampus (Redish et al., 2000) were observed when the animal was engaged in active exploration (although the possibility of an inertia-like effect during rest was not investigated in this case). Therefore, during active exploration, when acetylcholine levels are high and when we hypothesize Hebbian learning would occur, the presence of inertia in MTL neural firing rates should promote a build-up of correlations between MTL representations for neighboring viewpoints. This, in turn, should result in slowly varying representations and potentially encourage orientation invariance. The details of our implementation of temporal inertia are described below.

3.1.3 Egocentric Representation. Egocentric representations of the environments used for simulations 2 and 3 were derived using a method based on the boundary vector cell model of Hartley, Burgess, Lever, Cacucci, and O'Keefe (2000). To illustrate the exact nature of these representations, we first consider the situation depicted in Figure 5. Here, environmental boundaries are represented by the gray "walls," while the model's current location and head direction are represented by the black line segment. We discretize the boundaries into a set of line segments with a resolution of 10 segments per unit length. The boundary segments that are visible from the model's current location are shown in the egocentric reference frame, in which the model always faces along the positive y-axis, in the top panel of Figure 6 as open circles. This information can be transformed into a pattern of neural firing probabilities by forming a one-to-one correspondence between the set of egocentric layer neurons and a set of points on a radial grid centered at the model's current location, shown as filled circles in the top panel of Figure 6. The firing probability of the ith neuron due to the presence of the jth boundary segment is a gaussian function of the distance between the boundary segment location and the neuron's preferred location;

$$p_i^{Ego} = \frac{1}{r_j} e^{-(r_i^0 - r_j)^2 - \frac{(\theta_i^0 - \theta_j)^2}{2(\pi/20)^2}}, \tag{3.3}$$

where p_i^{Ego} is the firing probability for the ith egocentric neuron when driven by sensory input, (r_j, θ_j) are the radial coordinates of the jth boundary segment in the egocentric reference frame, and (r_i^0, θ_i^0) are the coordinates of a location where a boundary segment would have to be in order to elicit maximal firing from the ith egocentric neuron. The total

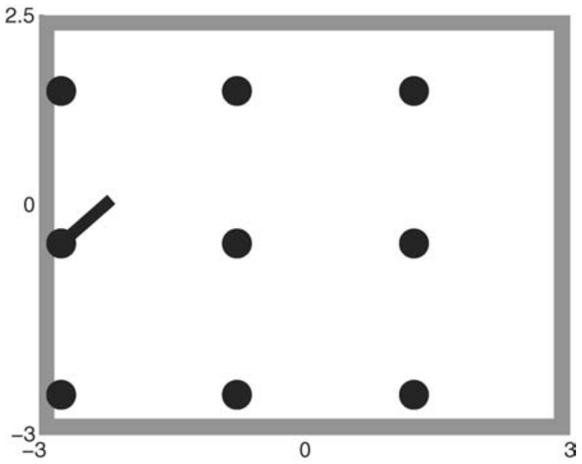

Figure 5: Schematic of a square environment. Thick gray lines represent boundaries (walls, for example), and filled black circles represent possible locations for the model. The black line segment extending from one location indicates that the model is at that location facing the direction in which the bar extends. Note: The possible model locations are offset slightly from center because they were determined automatically by an algorithm designed to accomplish this task for environments of arbitrary shape and size. There is no reason to expect this to affect the results obtained here in any way.

firing probability of the ith egocentric layer neuron due to the perceived environmental geometry is calculated by summing the contributions of all boundary segments to a maximum value of one, The information depicted in Figure 5 and the top panel of Figure 6 is shown in the bottom panel of the latter figure as neural firing probabilities calculated from equation 3.3.

3.1.4 Training Procedure. For this simulation we trained our transformation circuit on the linear track environment shown in Figure 7. Strong inhibitory connections from the head direction cell layer to the transformation layer were assumed to preexist. Accordingly, these weights were set so as to divide the transformation layer into as many equal-sized functional subsets as there are head directions, with one unique subset corresponding to each head direction. Thus, when the model takes on one of the available head directions, activity from the corresponding head direction cell strongly inhibits all activity in the transformation layer except for that of the appropriate subset. We trained the two layers of weights (egocentric-to-transformation and transformation-to-MTL layer weights)

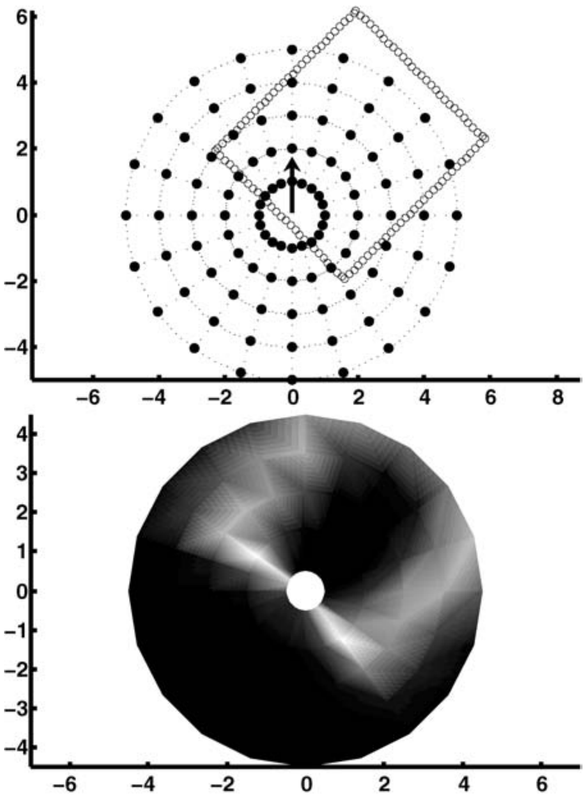

Figure 6: (Top) Open circles represent boundary segments that can be viewed from the model's location in Figure 5. These segments are shown in the model's egocentric reference frame in which the model itself is always located at the origin facing along the positive y-axis, as indicated by the arrow. Each filled circle on the radial grid corresponds to a neuron in the egocentric layer and is plotted at the position where a boundary segment would have to be in order to elicit maximal firing from the corresponding neuron. (Bottom) Relative firing probabilities of the egocentric layer neurons plotted at their corresponding grid points for the boundary configuration shown in the top panel.

in two separate navigation phases, treating the model as two stacked RBMs as described by Hinton et al. (2006). During the first exposure to the environment, the egocentric-to-transformation layer connections were trained using the RBM rules, with the egocentric layer representation of the current viewpoint acting as the visible layer data vector and the transformation layer acting as the hidden layer. For convenience, we used the online version of contrastive Hebbian learning described above, performing both a

Hebbian and an anti-Hebbian update of the weights at each step in the navigation. However, it is possible that the brain might employ these two learning phases in separate blocks, with Hebbian updates occurring during active exploration and anti-Hebbian updates occurring during quiet wakefulness or non-REM sleep. During the second exposure to the environment, the egocentric-to-transformation layer weights were kept fixed so that the transformation layer activations acted as fixed data vectors for training the transformation-to-MTL weights using the RBM rules, with the transformation layer acting as the visible layer and the MTL layer acting as the hidden layer. Training then proceeded in exactly the same way as for the egocentric-to-transformation layer weights, except that the effect of temporal inertia in the MTL neurons was added as described below.

While training the second half of the stacked RBM (transformation-to-MTL layer weights), we simulated temporal inertia by modulating the MTL neural activation function given by equation 3.1. In particular, we reduced the Hebbian phase firing probability of a MTL neuron at a given navigation step if it did not fire during the Hebbian phase at the previous navigation step. That is, if $p_j^M(t_i^{\mathrm{Hebb}})$ was the firing probability of the jth MTL neuron during the Hebbian phase at time t_i^{Hebb}, then we reset the firing probability of that neuron to

$$p_j'^M\left(t_i^{\mathrm{Hebb}}\right) = c \times p_j^M\left(t_i^{\mathrm{Hebb}}\right), \tag{3.4}$$

where c is a real-valued parameter slightly less than one, if and only if the jth MTL neuron did not fire during the Hebbian phase at time t_{i-1}^{Hebb}. In all other cases, p' was set equal to p. Exact values used for c are given below. Since c always had a value of one or very close to one, the overall weight update vector arising from any one navigation step was altered little by the effects of inertia, and the statistical interpretation of the RBM learning rules remained approximately valid, as evidenced by the drop in reconstruction error as training proceeded (see section 4.2). In section 5, we suggest two alternative mechanisms that could constitute physiological bases for temporal inertia. Note that we have modeled temporal inertia here as an inertia of inactivity by decreasing a neuron's firing probability if it did not fire at the previous step. The experimental evidence discussed above is more consistent with an inertia of activity that would result from an increased firing probability if the neuron did fire on the previous step. In fact, we have simulated both types of inertia separately and in combination. In all three cases, the results for our major measure of orientation invariance, D_M, were nearly identical. The advantage of modeling inertia as we have done here is that it results in sparser MTL representations and place-cell-like activity, at least in the case of the linear track (see below).

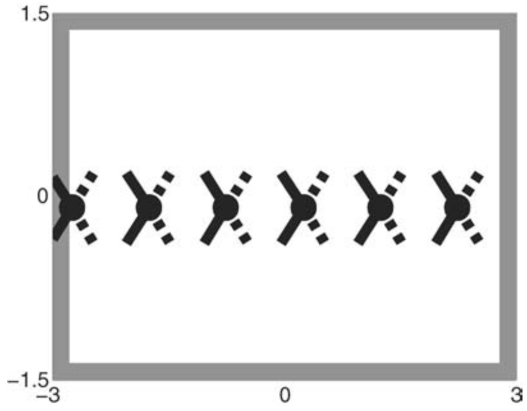

Figure 7: Schematic of the linear track environment. Thick gray lines represent environmental boundaries. Each possible model location is represented by a filled black circle, and each possible head direction at a given location is represented by either a solid or dashed line segment. Rightward headings are displayed as dashed lines and leftward headings as solid lines.

3.1.5 Environment and Navigation. The first environment on which we trained our model is shown in Figure 7. This "linear track" environment consisted of four walls enclosing a series of six locations. At each location were four possible head directions, which could be divided into two rightward headings (the two that have positive components in the rightward direction) and two leftward headings. Navigation was simulated in a manner roughly analogous to the way a rat explores a novel environment, alternatingly moving to a new location and making head turns. During training, navigation consisted of the model traversing the environment from the left-most position all the way to the right-most position and back again by making alternating locomotion steps and head turns. Before any given navigational step, if the model's current head direction was rightward, then it would translate one step rightward and randomly move its head to one of the two rightward directions, and then to the other rightward direction. Leftward steps were defined similarly.

Binary egocentric inputs for each viewpoint were calculated for each grid point on a radial grid covering the environment using the procedure described above. Since this procedure leads to continuous neural firing probabilities ranging from 0 to 1, any value greater than 0.2 was set to unity, while the remaining values were set to zero. The angular resolution of the egocentric grid was chosen to be $\pi/10$ radians, while the radial resolution was chosen to be unity, with values ranging from 1 to 6, for a total of $20 \times 6 = 120$ egocentric neurons. The number of neurons in the transformation layer was chosen to be 4 (head directions) $\times 120 = 480$, while the number of MTL neurons was chosen to be 120. Inhibitory weights from the head direction system divided the transformation layer into four competing pools of 120 neurons each, with each subset active for one unique head direction only. These values were chosen because our previous work (Byrne et al., 2007) demonstrates that a network with these interlayer size ratios can perform egocentric-allocentric transformations accurately.

In the first training phase, the connection weights between the egocentric input layer and the transformation layer were trained so that the model could reconstruct egocentric input from the resulting transformation layer representations. This training was performed using the standard RBM learning rules with $\eta = 0.05$ and all initial weights set to zero. A larger learning rate parameter was tested in preliminary simulations, but a smaller value tended to lead to more stable final transformation layer firing representations over repeated sampling, making it easier for the MTL layer to learn representations of transformation layer activity. In the second phase of training, the transformation-to-MTL weights were learned using the same RBM procedure, with c set to 0.9 for the MTL neurons. Again, the initial weights were set to zero. For this phase, η was set to the smaller value of 0.001 because larger learning rates were found to generate MTL representations with lower levels of orientation invariance.

3.2 Results and Discussion. Reconstruction error was defined as

$$\sqrt{\frac{1}{N_v N}\sum_{i=1}^{N}[E(x_i) - \bar{E}_r(x_i)]^2}, \tag{3.5}$$

where N_v is the number of egocentric neurons, N is the number of viewpoints within the environment, and $\bar{E}_r$ is the egocentric representation reconstruction averaged over 500 samples. For the first phase of training, in which reconstructions of data were generated using only the corresponding transformation layer representations, reconstruction error is plotted in the upper-left panel of Figure 8. Notice that the model learned to reconstruct egocentric inputs from their transformation layer representations after about 2.5×10^5 navigation steps. However, training was continued for an extended period to further reduce variance in the transformation layer representations. The number of training steps used here might seem excessive, but it should be noted that our model was trained de novo, whereas a real neural system pretrained on a large ensemble of environments could learn the mapping for a new environment much more quickly. We return to this point in section 5.

In the second training phase, reconstruction error was calculated again using equation 3.5, but in this case reconstruction relied on the entire circuit, with egocentric input feeding forward to the MTL layer via the transformation layer and then back through the entire circuit to perform the requisite reconstruction. Results averaged over five networks are plotted

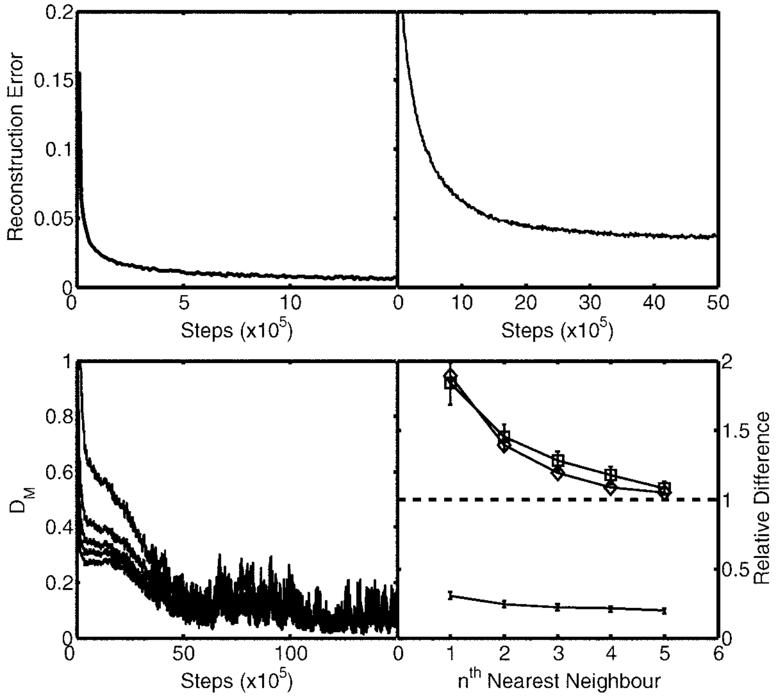

Figure 8: (Upper left) Reconstruction error as a function of training step for the linear track environment during training of the egocentric-to-transformation weights. (Upper right) Reconstruction error as a function of training step for the linear track environment during training of the transformation-to-MTL weights. (Lower left) A family of D_M values for the linear track environment plotted against training step, with each curve corresponding to the orientation invariance measure for a given spatial separation. The upper curve is $D_M(1)$, and the lower curve is $D_M(5)$, that is, the orientation dependence of MTL representations at spatial separations of 1 to 5. (Lower right) Orientation-invariance results for the model trained with the RBM rules on the linear track environment. The diamonds are $D_E(n)$. The squares represent the same calculation for MTL representations (i.e., $D_M(n)$), but with only rotations between leftward and rightward directions considered. The lower curve with error bars alone represents $D_M(n)$, but with only rotations between leftward and leftward directions or rightward and rightward directions considered. Error bars represent standard errors of the means calculated across five networks.

in the upper-right panel of Figure 8. The full model learned to reconstruct the data reasonably well after 15×10^5 steps, but because of the effects of the temporal inertia, it did not perform as well at reconstruction as the simple egocentric layer–transformation layer circuit alone. As in the first

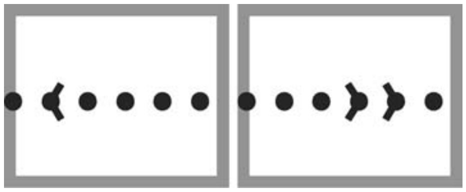

Figure 9: Examples of two linear track model place cells. The solid line segments indicate the viewpoints at which the given cell responds.

simulation, orientation invariance of MTL representations was measured using equation 2.3. However, whereas the MTL activation vector for a given location and orientation (i.e., $M(x_i)$) was determined uniquely by the egocentric input via deterministic equations (neurons were rate-coded) in the first simulation, here its mean value over 500 samples was used in the determination $D_M(n)$. Also, because the model switched only between leftward and rightward directions at the end points of the track, $D_M(n)$ was calculated separately for head rotations between two same-direction traversals (both rightward or both leftward) and head rotations between a leftward and rightward direction. It should be noted that in the former case, $D_M(n)$ drops rapidly during early training but then begins to fluctuate, followed by further small drops after extended periods of time. This is illustrated in the bottom left panel of Figure 8. Since convergence of this quantity is therefore difficult to determine, we let training simulations run for approximately 50×10^5 steps, after which reconstruction error tended to have reached a minimum and $D_M(n)$ had gone through its first major drop in value. Values of $D_M(n)$ and $D_E(n)$ averaged over five networks are shown in the bottom right panel of Figure 8. Clearly, MTL representations tend toward orientation invariance if only rightward or leftward travel directions are considered in isolation.

From the bottom right panel of Figure 8, we see that the network exhibited orientation invariance within one direction of travel but not between travel directions. This is consistent with the fact that place cells tend to show directionality in their firing on the linear track (McNaughton, Barnes, & O'Keefe, 1983). To examine this further, we analyzed the response properties of our model MTL neurons to see if they exhibited place-cell-like behavior. First, a threshold was defined such that if a neuron fired at a given viewpoint in more than 75% of 500 samples, then it was considered to respond at that viewpoint. Second, if that neuron responded in a sufficiently localized way, defined here as firing at one location or two neighboring locations, for both leftward or both rightward directions, but at no additional viewpoints, then it was labeled as a directionally selective place cell. Examples of such cells are shown in Figure 9. The results of this analysis averaged over five networks indicate that $19 \pm 4\%$ (mean $\pm$

Standard error of the mean (S.E.M.) calculated over five networks) of MTL neurons responded for at least one viewpoint on the track, while at any given viewpoint, an average of $5.3 \pm 0.5\%$ of neurons responded. Of responding neurons, $45 \pm 2\%$ were directionally selective place cells. An analysis of the firing properties of transformation layer neurons revealed no discernible patterns other than that the neurons were selective for head direction, as built into the network's construction.

Results for simulations in which an "inertia of activity" was employed (not shown here) were nearly identical to the above results in terms of the measure of orientation invariance, D_M. However, MTL neurons in these latter simulations tended to show firing opposite to those described above. That is, instead of demonstrating spatially localized, orientation-invariant firing fields, MTL neurons in these simulations showed firing in most locations, but with spatially localized, orientation-invariant fields of inactivity.

It should be noted that Battaglia, Sutherland, and McNaughton (2004) have found that when objects are added to linear tracks, thus adding visual complexity, place cell firing becomes direction independent, as in the open field. The direction dependence of our linear track place fields arises because when the model travels in leftward or rightward directions, it always looks in directions that are predominantly leftward or rightward. We simulated navigation this way because we assumed that an animal traveling along such a track would focus its attention in a predominantly forward direction. Adding complexity to the track might affect how it is explored or attended to by the animal, perhaps even causing it to be treated more like a full two-dimensional environment.

4 Simulation 3: RBM Method Applied to Small Square Environment

4.1 Methods. The second environment on which we trained our model is shown in Figure 10. This "box" environment consisted of four walls enclosing a collection of nine locations. In this environment, the model could take on any one of four equally spaced head directions at each location. During training, navigation consisted of the model taking alternating translation and rotation steps to form a random trajectory through the environment. The model would move to a nearest-neighbor location within π radians of its current head direction while maintaining this head direction, and then randomly make either a clockwise or counterclockwise turn through one head direction increment ($\pi/2$ radians).

Binary egocentric inputs for each viewpoint were calculated at points on a radial grid covering the environment as in the previous simulation. The angular resolution of the egocentric grid was $\pi/10$ radians, while the radial resolution was unity, with values ranging from 1 to 5 for a total of $20 \times 5 = 100$ egocentric neurons. The number of neurons in the transformation layer was 400 (4 head directions $\times 100$ egocentric inputs), while the number of MTL neurons was 100. Inhibitory weights from the head direction system

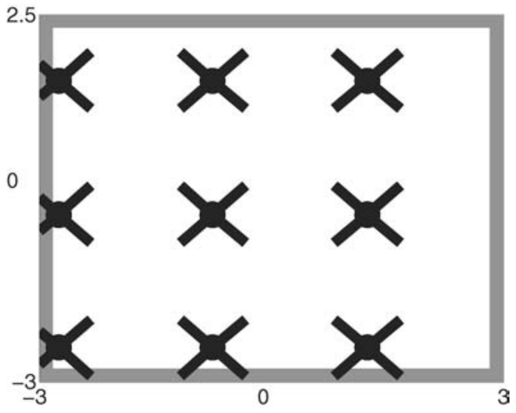

Figure 10: Schematic of the square environment. Thick gray lines represent environmental boundaries. Each possible model location is represented by a filled black circle, while each possible head direction at a given location is represented by a solid line segment.

divided the transformation layer into four competing pools of 100 neurons with one active for each head direction.

The remaining procedures and parameters were identical to those used in simulation 2, except that c was set equal to 0.8 in order to generate slightly stronger inertia, which was found to give slightly better results. Training of the egocentric-to-transformation weights was performed for 6.5×10^5 steps, while training for the transformation-to-MTL weights was performed for 20×10^5 steps.

4.2 Results and Discussion. Unlike the previous environments, the maximum linear dimension of the box environment was rather short, so $D_M(n)$ and $D_E(n)$ could be calculated only for $n = 1, 2,$ and 3. These quantities are plotted in Figure 11, with $D_M(n)$ averaged over five independently trained networks. Again $D_M(n)$ is small compared with one as well as with $D_E(n)$, indicating the development of orientation invariance in MTL representations. An analysis of MTL neuron responses indicated sparse coding, with $25.2 \pm 3.8\%$ (mean $\pm$ SEM calculated over five networks) of neurons responding at at least one viewpoint within the environment, but with only $4.3 \pm 0.3\%$ responding at any one viewpoint.

In the open field, a place cell fires selectively for some localized region of the environment and shows little dependence on head direction (Muller et al., 1994). While some of the MTL neurons in our trained models did respond selectively for one or two neighboring locations only, and did so for all head directions, the majority were difficult to classify. This may simply have been because our square environment did not contain a large

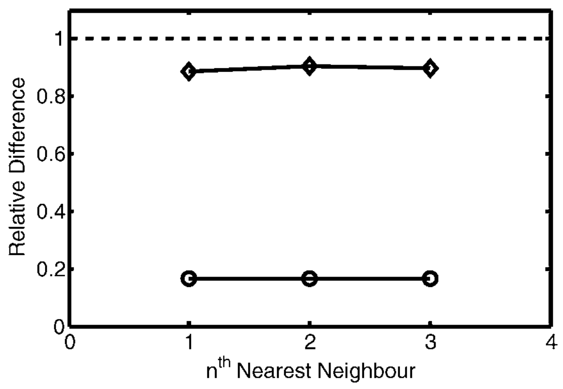

Figure 11: Orientation-invariance results for the model trained with the RBM rules on the square environment. The diamonds are $D_E(n)$, while the circles are $D_M(n)$ averaged over five networks (error bars are too small to plot here).

enough number of exploration locations to provide a good measure of how "localized" a given neuron's response area was. In addition to the few observed place cell responses (approximately 20% of responding MTL neurons), we also found that some model neurons tended to respond at a fixed distance from one of the walls for the majority of head directions. Such responses resembled those of boundary vector cells, a type of neuron that has been suggested by O'Keefe and Burgess (1996) and Hartley et al. (2000) to drive the firing of place cells. Together, cells that responded in a way qualitatively similar to place cells or boundary vector cells made up about half of responding MTL neurons.

As with the linear track simulations of the previous section, switching to an inertia of activity in these simulations was found to have no appreciable effect on the orientation-invariance measure, D_M.

5 General Discussion

In this letter, we have argued that two principles are responsible for learning orientation-invariant representations of environments: (1) the need to form accurately retrievable, long-term memory representations of spatial layouts and (2) a temporal inertia constraint within MTL memory structures. Our initial approach was to train a neural network model to minimize a cost function embodying these principles. Although this approach showed that the principles are indeed consistent with the learning of orientation invariant representations, it suffered from two shortcomings: the unbiological nature of the learning rules and the existence of numerous local minima in the cost function. These problems were overcome by the second approach, using an RBM trained by correlational weight updates; such weight updates might be accomplished respectively by long-term potentiation and depression in the brain. The RBM also generated MTL representations that tended to orientation invariance on the whole and also exhibited several characteristics reminiscent of place cells found in the mammalian hippocampus, at least for the linear track environment. Abstract environments with a higher density of exploration points should be trained to clarify the response properties of model MTL neurons in an open-field setting.

There are several ways in which the simple form of temporal inertia we have proposed might be accomplished by neural systems. For example, gain modulation, which has been observed in posterior parietal cortex (Snyder et al., 1998) and modeled with multiplicative synapses (Pouget & Sejnowski, 1997), could allow self-connected neurons to modulate their future sensitivity to afferent input. Alternatively, it could be the case that during a Hebbian phase, transformation neurons alone cannot drive MTL neurons to saturation. Instead, if a given MTL neuron fired at one navigational step and continued receiving high levels of input at the next step, then voltage-dependent N-methyl-D-aspartate receptors with slow dynamics (Jahr & Stevens, 1990) might begin to contribute to the driving postsynaptic currents, thus allowing saturation on the next step. Of course, these two mechanisms are more consistent with an inertia of activity than the inertia of inactivity that we have concentrated on here. However, as we have described above, employing the former kind of inertia generates model MTL neuron responses that consist of localized fields of orientation-invariant inactivity. Such responses could be easily inverted through a layer of inhibitory interneurons to give place-cell-like behavior.

Although the RBM overcame some of the shortcomings of the initial model, it required an excessively large number of training iterations. However, the large number of steps might not be as unrealistic as it first seems. For example, a small cluster of biological neurons, perhaps corresponding to one neural unit in our model, could be expected to produce tens or hundreds of spikes per second if highly excited. A quick calculation would reveal that a few hours of active exploration each day for a number of days or weeks could reasonably be expected to generate such a large number of spikes. Thus, if an animal experienced multiple environments during its early development, a set of weights that would generate place-cell-like characteristics in both familiar and novel environments might be learned. After this period, only fine-tuning would be required to encode novel spatial layouts. Indeed, when rats are placed in novel enclosures, they show somewhat distorted place cell firing immediately (O'Keefe & Burgess, 1996). Slight refinement of connection strengths might be all that is required for them to hone this representation and learn their novel surroundings (Barry et al., 2006). In terms of future work, it would be beneficial if a number of unique environments were constructed and the model trained on some subset of them simultaneously. We predict that after such training, subsequent learning would be rapid, and orientation-invariant representations of the remaining environments would be learned quickly. This process would require larger networks, more complex environments, and more training time.

Recently, Rolls and Kesner (2006) have shown that place cell responses can be generated easily by upstream grid cells. However, to our knowledge, no models presented prior to the discovery of grid cells have relied on units with such firing properties, nor have any such models demonstrated the emergence of grid cell responses over the course of learning. Similarly, although our model does give rise to place-cell-like responses under certain circumstances, it cannot be considered a full model of the system that transforms egocentric information into place cell behavior. This leaves at least two possible interpretations for our model. First, grid cell activity might not constitute the first level of orientation-invariant spatial representation in the brain. In this case, our model MTL layer might be related to some upstream driver of grid cell activity. The second possibility is that medial temporal neurons in our model are displaying grid cell firing patterns, but that the small size of the simulated environments does not allow more than one vertex to be seen in the response of a given cell. This would require an individual MTL neuron to receive strong input not only when the model is presented with one particular egocentric view, but also when the model is presented with a view that results from pure translation to any other vertex location. Although an RBM tends to learn a unique hidden state for each unique visible state, the possibility of an MTL neuron being responsive to multiple, distinct views is not problematic so long as the population state vector is unique at each location. However, the likelihood of our model learning MTL representations with this level of structure still seems low without the addition of further constraints. O'Keefe and Burgess (2005) suggest one such constraint that could lead to the emergence of grid cells: that a gridlike interference pattern is generated by the interaction between theta-frequency-modulated septal inputs to the hippocampus and near-theta-frequency subthreshold oscillations in membrane potential in entorhinal cells.

With weights in the transformation model set to completely random values, we would expect rapid variation in egocentric input generally to produce rapid variation in MTL firing. However, after training, MTL representations were found to vary quite slowly during rotation. The temporal inertia employed for the RBM simulations above has its largest slowing effect on the firing of a given MTL neuron when the activity of that neuron varies the fastest. Therefore, the network learns rotational invariance in part by finding slowly varying representations of more quickly varying input. This is similar in spirit to the neural model of Wiskott and Sejnowski (2002), which learns such invariances by explicitly finding outputs that vary as slowly as possible in time. As with many models of place cell learning, though, this model does not naturally learn the inverse transformation needed to recreate the full input.

As a final point, we offer a possible route to testing our ideas experimentally. Simulation results for both the RBM trained in the square and linear environments and for the model trained with the original cost function in the very simple linear environment indicate that D_M, the measure of orientation invariance in MTL representations, continued to decrease (indicating increasing orientation invariance) well beyond the point at which the model learned to accurately reconstruct egocentric representations. It would therefore be of great interest if such an effect could be found in animals, perhaps during early development. Martin and Berthoz (2002) have shown that well-defined head direction cell activity develops in young rats before place cell activity becomes stable, a requirement for our model, and that this place cell stability is not well established until after 50 days of age. In order to test our prediction, experiments that attempt to correlate directly the development of place cell activity with behavioral performance must be conducted.

Acknowledgments

This research was supported by a grant from the Natural Sciences and Engineering Research Council of Canada to S.B.

References

Ackley, D., Hinton, G., & Sejnowski, T. (1985). A learning algorithm for Boltzmann machines. *Cognitive Science*, *9*(1), 147–169.

Barnes, C. (1988). Spatial learning and memory processes: The search for their neurobiological mechanisms in the rat. *Trends Neurosci.*, *11*(4), 163–169.

Barry, C., Lever, C., Hayman, R., Hartley, T., Burton, S., O'Keefe, J., et al. (2006). The boundary vector cell model of place cell firing and spatial memory. *Reviews in the Neurosciences*, *17*(1–2), 71–97.

Battaglia, F. P., Sutherland, G. R., & McNaughton, B. L. (2004). Local sensory cues and place cell directionality: Additional evidence of prospective coding in the hippocampus. *Journal of Neuroscience*, *24*, 4541–4550.

Becker, S., & Burgess, N. (2001). A model of spatial recall, mental imagery and neglect. In T. Leen, T. Ditterich, & V. Tresp (Eds.), *Advances in neural information processing systems*, 13 (pp. 96–102). Cambridge, MA: MIT Press.

Bohbot, V., Kalina, M., Stepankova, K., Spackova, N., Petrides, M., & Nadel, L. (1998). Spatial memory deficits in patients with lesions to the right hippocampus and to the right parahippocampal cortex. *Neuropsychologica*, *36*(11), 1217–1238.

Burgess, N., Becker, S., King, J., & O'Keefe, J. (2001). Memory for events and their spatial context: Models and experiments. *Philos. Trans. R. Soc. Lond. B Biol. Sci.*, *356*(1413), 1493–1503.

Burgess, N., Maguire, E., & O'Keefe, J. (2002). The human hippocampus and spatial and episodic memory. *Neuron*, *35*, 625–641.

Byrne, P., & Becker, S. (2004). Modelling mental navigation in scenes with multiple objects. *Neural Computation*, *16*, 1851–1872.

Byrne, P., Becker, S., & Burgess, N. (2007). Remembering the past and imagining the future: A neural model of spatial memory and imagery. *Psych. Rev., 114*, 340–375

Chafee, M., & Goldman-Rakic, P. (1998). Matching patterns of activity in primate prefrontal area 8a and parietal area 7ip neurons during a spatial working memory task. *Journal of Neurophysiology*, *79*(6), 2919–2940.

Crane, J., & Milner, B. (2005). What went where? Impaired object-location learning in patients with right hippocampal lesions. *Hippocampus, 15*, 216–231.

Deneve, S., Latham, P., & Pouget, A. (2001). Efficient computation and cue integration with noisy population codes. *Nature Neurosci.*, *4*, 826–831.

Egorov, A., Hanmam, B., Fransen, E., Hasselmo, M., & Alonso, A. (2002). Graded persistent activity in entorhinal cortex neurons. *Nature*, *420*, 173–178.

Ekstrom, A., Kahana, M., Caplan, J., Fields, T., Isham, E., Newman, E., et al. (2003). Cellular networks underlying human spatial navigation. *Nature*, *425*, 184–187.

Foster, D. J., & Wilson, M. A. (2006). Reverse replay of behavioural sequences in hippocampal place cells during the awake state. *Nature*, *440*, 680–683.

Funahashi, S., Bruce, C., & Goldman-Rakic, P. (1989). Mnemonic coding of visual space in the monkey's dorsolateral prefrontal cortex. *Journal of Neurophysiology*, *61*(2), 331–348.

Hafting, T., Fyhn, M., Molden, S., Moser, M. B., & Moser, E. (2005). Microstructure of a spatial map in the entorhinal cortex. *Nature*, *436*(7052), 801–806.

Hartley, T., Burgess, N., Lever, C., Cacucci,F., & O'Keefe, J. (2000). Modelling place fields in terms of the cortical inputs to the hippocampus. *Hippocampus*, *10*(4), 369–379.

Hasselmo, M. (1999). Neuromodulation: Acetylcholine and memory consolidation. *Trends Cogn. Sci.*, *3*(9), 351–359.

Hasselmo, M. E., & McGaughy, J. (2004). High acetylcholine levels set circuit dynamics for attention and encoding and low acetylcholine levels set dynamics for consolidation. *Progress in Brain Research*, *145*, 207–231.

Hinton, G. (2002). Training products of experts by minimizing contrastive divergence. *Neural Computation*, *14*(8), 1771–1800.

Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. *Neural Comput.*, *18*(7), 1527–1554.

Jahr, C. E., & Stevens, C. F. (1990). Voltage dependence of NMDA-activated macroscopic conductances predicted by single-channel kinetics. *J. Neurosci.*, *10*, 3178–3182.

Jarrard, L. (1993). On the role of the hippocampus in learning and memory in the rat. *Behav. Neural Biol.*, *60*, 9–26.

Kali, S., & Dayan, P. (2000). The involvement of recurrent connections in area CA3 in establishing the properties of place fields: A model. *Journal of Neuroscience*, *20*(19), 7463–7477.

King, J., Burgess, N., Hartley, T., Vargha-Khadem, F., & O'Keefe, J. (2002). Human hippocampus and viewpoint dependence in spatial memory. *Hippocampus*, *12*(6), 811–820.

Kosko, B. (1988). Bidirectional associative memories. *IEEE Transactions on Systems, Man and Cybernetics*, *18*, 49–60.

LeCun, Y. (1985). A learning scheme for asymmetric threshold network. In CESTA-AFCET (Ed.), *Cognitiva 85: A la Frontière de l'Intelligence Artificielle des Sciences de la Connaissance des Neurosciences* (pp. 599–604), Paris.

Martin, P. D., & Berthoz, A. (2002). Development of spatial firing in the hippocampus of young rats. *Hippocampus*, *12*, 465–480.

Matsumura, N., Nishijo, H., Tamura, R., Eifuku, S., Endo, S., & Ono, T. (1999). Spatial- and task-dependent neuronal responses during real and virtual translocation in the monkey hippocampal formation. *J. Neurosci.*, *19*(6), 2381–2393.

Mazzoni, P., Andersen, R., & Jordan, M. (1991). A more biologically plausible learning rule than backpropagation applied to a network model of cortical area 7a. *Cerebral Cortex*, *1*, 293–307.

McNaughton, B., Barnes, C., & O'Keefe, J. (1983). The contributions of position, direction, and velocity to single unit activity in the hippocampus of freely-moving rats. *Exp. Brain Res.*, *52*(1), 41–49.

Milner, A., Paulignan, Y., Dijkerman, H., Michel, F., & Jeannerod, M. (1999). A paradoxical improvement of misreaching in optic ataxia: New evidence for two separate neural systems for visual localization. *Proc. Royal Soc. Lon. B Biol. Sci.*, *266*(1434), 2225–2229.

Morris, R., Garrard, P., Rawlins, J., & O'Keefe, J. (1982). Place navigation impaired in rats with hippocampal lesions. *Nature*, *297*, 681–683.

Muller, R. U., Bostock, E., Taube, J. S., & Kubie, J. L. (1994). On the directional firing properties of hippocampal place cells. *J. Neurosci.*, *4*(12), 7235–7251.

O'Keefe, J. (1976). Place units in the hippocampus of the freely moving rat. *Exp. Neurol.*, *51*(1), 78–109.

O'Keefe, J., & Burgess, N. (1996). Geometric determinants of the place fields of hippocampal neurons. *Nature*, *381*, 425–428.

O'Keefe, J., & Burgess, N. (2005). Dual phase and rate coding in hippocampal place cells: Theoretical significance and relationship to entorhinal grid cells. *Hippocampus*, *15*(7), 853–866.

Ono, T., Nakamura, K., Nishijo, H., & Eifuku, S. (1993). Monkey hippocampal neurons related to spatial and nonspatial functions. *Journal of Neurophysiology*, *70*(4), 1516–1529.

Parker, D. (1985). *Learning-logic.* (Tech. Rep. 1). Cambridge, MA: MIT Center for Computational Research in Economics and Management Science.

Pouget, A., & Sejnowski, T. (1997). Spatial transformations in the parietal cortex using basis functions. *J. Cogn. Neurosci.*, *9*, 222–237.

Redish, A., McNaughton, B., & Barnes, C. (2000). Place cell firing shows an inertia-like process. *Neurocomputing*, *32–33*, 235–241.

Riedmiller, M., & Braun, H. (1993). A direct adaptive method for faster backpropagation learning: The RPROP algorithm. In *Proceedings of the International Conference on Neural Networks* (pp. 586–591). San Francisco: IEEE.

Rolls, E. T., & Kesner, R. P. (2006). A computational theory of hippocampal function, and empirical tests of the theory. *Progress in Neurobiology*, *79*(1), 1–48.

Rolls, E., & O'Mara, S. (1995). View-responsive neurons in the primate hippocampal complex. *Hippocampus*, *5*(5), 409–424.

Rumelhart, D., Hinton, G., & Williams, R. (1986). Learning internal representations by error propagation. In D. E. Rumelhart & J. L. McClelland (Eds.), *Parallel distributed* *processing: Explorations in the microstructure of cognition* (Vol. 1). Cambridge, MA: MIT Press.

Salinas, E., & Abbott, L. (1996). A model of multiplicative neural responses in parietal cortex. *Proc. Natl. Acad. Sci. USA*, *93*, 11956–11961.

Sargolini, F., Fyhn, M., Hafting, T., McNaughton, B. L., Witter, M. P., Moser, M. B., et al. (2006). Conjunctive representation of position, direction, and velocity in entorhinal cortex. *Science*, *312*(5774), 758–762.

Sharp, P. E. (1991) Computer simulation of hippocampal place cells. *Psychobiology*, *19*(2), 103–115.

Smolensky, P. (1986).In formation processing in dynamical systems: Foundations of harmony theory. In D. E. Rumelhart, & J. L. McClelland (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition* (Vol. 1). Cambridge, MA: MIT Press.

Snyder, L., Grieve, K., Brotchie, P., & Andersen, R. (1998). Separate body- and world-referenced representations of visual spacein parietal cortex. *Nature*, *394*(6696), 887–891.

Taube, J. (1998). Head direction cells and the neurophysiological basis for a sense of direction.*Prog. Neurobiol.*, *55*(3), 225–256.

Trullier, O., Wiener, S. I., Berthoz, A., & Meyer, J. A. (1997). Biologically based artificial navigation systems: Review and prospects. *Progress in Neurobiology*, *51*, 483–544.

Werbos, P. J. (1974). *Beyond regression: New tools for prediction and analysis in the behavioral sciences.* Unpublished doctoral dissertation, Harvard University.

Wiskott, L., & Sejnowski, T. J. (2002). Slow feature analysis: Unsupervised learning of invariances. *Neural Computation*, *14*, 715–770.

Xing, J., & Andersen, R. (2000). Models of the posterior parietal cortex which perform multimodal integration and represent space in several coordinate frames. *J. Cogn. Neurosci.*, *12*, 601–614.

Zipser, D., & Andersen, R. (1988). A back-propagation programmed network that simulates response properties of a subset of posterior parietal neurons. *Nature*, *331*, 679–684.

Received October 10, 2006; accepted April 5, 2007.